83. The computer method of Claim 82 further including the acts of defining a character string size having a predetermined number of characters to be in the abbreviated character string;

at the completion of each of the act C_1, C_2 and C_3 comparing a remaining portion of the character string with the character string size and terminating character removal if the remaining portion of the character string equals the character string size.

84. The method of Claim 43 wherein the abbreviating step further includes the steps of sequentially removing predetermined individual characters until the at least one entry has a width equivalent to a predetermined number of characteristics.

REMARKS

This amendment is in response to the Office Action mailed July 17, 2001.

Claim 42 is rejected under 35 USC 112, first paragraph, as containing subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains or with which it is most nearly connected, to make and/or use the invention. To support this rejection the examiner seems to use Claim 42 to derive the abbreviation set forth in Fig. 8.

In response, this approach seemed flawed in that Fig. 8 is a result of applying the invention (abbreviation) set forth in Fig. 7 and the appropriate portion of the specification rather than Claim 42. A rejection under 35 USC 112, first paragraph, requires that the teaching in the claim is supported somewhere within the four corners of the document (i.e. patent application) rather than a specific portion which the examiner focuses on. We believe that Claim 42 is supported by Fig. 7 and the teaching in the related function of the specification.

The examiner's analysis further indicates what appears to be a misunderstanding of applicant's invention. To assist understanding of applicant's invention an example is traced through Fig. 7 which is the flowchart of applicant's invention. In this example the initial character string to abbreviate is:
"Disk 1 workload"

The string comes into block 169, and the first character type is selected. The first character type is space characters. Block 170 selects the character type and accordingly goes to block 172, which states that character type 1 (spaces) will be removed.

Block 180 refers to the last character, which is "d". It is important to note that we start at the last character because we are going to go backwards through the string, removing spaces.

Block 182 tests to see if the string is short enough yet. Since "Disk 1 Workload" is greater than 3 characters, the answer is no, so we continue to block 186.

Block 186 tests to see if the current character (the last character, "d") is of type 1 (a space). The answer is no, so we proceed to block 190 where we back up one character to the "a". Block 192 checks to see if we've reached the beginning of the string. Since we have not, we go back to block 182.

We repeat this loop several times, moving backward through the string. We skip over the "a" in block 186 because it is not a space. The next time through, we skip over the "o" because it is not a space. And so forth, until we skip over the "W" because it is not a space.

But the next time through the loop, we come to the space before "Workload." Block 186 determines that this is a space, and block 188 removes it from the string. Now we have:

"Disk 1Workload"

The string is shorter now, but still longer than 3 characters. So when we get to block 182 the next time through the loop, we keep going. We skip the "1" because it is not a space, but then we come to the space before the "1". It is detected by block 186 and removed by block 188. Now we have:

"Disk1Workload"

We continue looping, passing over the "k", then the "s", then the "i", and finally the "D". None of these characters are spaces, so none are removed.

After we pass over "D", block 192 determines that we have reached the beginning of the string. Hence, block 193 moves to the next character type, Type Two, which is lowercase vowels. Block 174 selects the test for lowercase vowels. Block 180 sets us back to the end of the string. Now we will pass backwards through the string once again, this time removing lowercase vowels.

Block 182 determines that the string is still longer than 3 characters, and block 186 tests "d" to see if it is a lowercase vowel It is not, so block 190 moves backward to the previous character. Next time through the loop, block 186 determines that the "a" is a lowercase vowel and block 186 removes it. Now we have:

"Disk1Worklod"

Block 190 backs up another character Block 192 determines we have not arrived at the beginning of the string, so it sends us to block 182. Block 182 determines that the string is still longer than 3 characters, so we continue. Block 186 tests the "o" and finds that it is a lowercase vowel. Block 188 removes it. Now we have:

"Disk1Workld"

Continuing to move backwards, we pass over the "l", the "k", and the "r" in the way defined above. However, block 186 determines that the "o" is a lowercase vowel (Type 2 character), so block 188 removes it, leaving:

"Disk1Wrkld"

The process continues moving backward through the string until it finds that the "i" is a lowercase vowel. Then we have:

"Dsk1Wrkld"

As with Type 1 characters (spaces), the search for Type 2 characters eventually arrives at the beginning of the string, "D". At this point, block 192 sends us to block 193, which goes to the next character type, lowercase characters. Block 180 sets us back to the end of the string. Then the loop begins working its way toward the beginning of the string, removing lowercase characters. The progression will look like this:

"Dsk1Wrkl"
"Dsk1Wrk"
"Dsk1Wr"
"Dsk1W"
"Ds1W"
"D1W"

When the "s" is removed, there are only 3 characters left. At this point, block 182 will find that the string is now abbreviated to the requested length. Hence, the process is completed.

The invention is the process of moving through a string, deleting characters of successive character types. Notice that each time the string gets shorter, it maintains good readability and uniqueness for the new length. Consider the progression:

"Disk 1 Workload"
"Disk 1Workload"
"Disk1Workload"
"Disk1Worklod"
"Disk1Workld"
"Disk1Wrkld"
"Dsk1Wrkld"
"Dsk1Wrkl"
"Dsk1Wrk"
"Dsk1Wr"

"Dsk1W"
"Ds1W"
"D1W"

If one has 10 characters to abbreviate "Disk 1 Workload", then "Disk1Wrkld" is arguably the most readable abbreviation. If one only has nine characters, then "Dsk1Wrkld" is as readable as possible. If one only has 3 characters, "D1W" is the best. The algorithm is an attempt to maintain readability when strings must be shortened, and predefined abbreviations are not available. This is clearly much better than the prior art of simple truncation.

The preferred embodiment uses three character types: spaces, lowercase vowels, and lowercase characters. Alternative embodiments could use a different set of character types. This allows support for various languages other than English, which may not even have vowels or spaces or lowercase letters. It might also be possible to improve the algorithm for English by adding more character types. But the basic invention is the same.

With respect to the rejection of Claim 42 the claim is amended to include the three character types.

Claims 46 and 47 are rejected under 35 USC 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. In this regard the examiner seemed to argue that it is confusing since a lower case character may include a lower case vowel.

In response, applicant fails to see any indefiniteness in these claims or understands the Examiner's argument supporting the rejection. Claim 46 recites the removal of lower case vowels. Claim 47 does not depend on Claim 46. Claim 47 calls for removing lower case characters. Claims 46 and 47 are removing "lower case vowels" and "lower case characters", respectively. Claim 47 does not depend on Claim 46; so applicant fails to see any indefiniteness in the claims. If the Examiner still persists with this rejection, applicant respectfully requests a better explanation as to why the claims are indefinite.

Claims 42-73 are rejected under 35 USC 103(a) as being unpatentable over the applicant-admitted prior art Fig. 4 of the specification. To support the rejection the examiner compared applicant's claim with what he erroneously concluded is shown in Fig. 4. The examiner then concluded that applicant's claim without any additional concrete evidence is obvious in view of the teaching of Fig. 4.

In response, applicant wishes to point out that the examiner's analysis seems flawed in that Fig. 4, the prior art, is derived by the truncation technique set forth in Fig. 3 and described on page 8 of applicant's specification. A review of the truncation technique set forth in applicant's specification clearly shows error in the examiner's analysis. In particular, what the examiner alleged Fig. 4 shows is nothing more than information gleaned from applicant's abbreviation technique disclosed in Fig. 7 and described in the appropriate section of applicant's specification.

If the Examiner is relying on prior art disclosed in applicant's specification the Examiner must also accept the shortcomings and problems associated with the prior art by applicant. The examiner cannot pick and choose only the portion of the prior art that he believes is favorable to his position. The prior art must be considered as a whole. In this regard applicant argues the acts including classification of characters that are removed from the string of characters to be abbreviated is not suggested in the prior art. In addition, the truncation method of the prior art has, among other things, the problem of losing vital information in the remaining portion of a truncated character string (applicant's specification, page 9, lines 8-20). It is applicant's position that the different acts (process steps set forth in applicant's claim) and the solution of the problem associated with truncation are clear evidence of unobviousness.

In addition, the benefits (see page 16, applicant's specification) set forth in applicant's specification and the difference in process steps between the prior art and applicant's claims are evidence of unobviousness.

Whereas the prior art uses truncation to derive Fig. 4 applicant uses a different technique (i.e. abbreviation) to shorten the character string and hence provide the desirable results set forth in Fig. 8 of applicant's specification. The distinctions between truncation and abbreviation are set forth in the argument in the preliminary amendment filed with this continuation. The argument is applicable and is incorporated herein by reference.

In addition to the different technique used in the prior art and applicant's invention applicant argues that the prior art is plagued with problem set forth on page 9, lines 8-20 of applicant's specification. It is applicant's position that because of the different technique resulting in the solution of the problem in the prior art the claims in applicant's invention are not obvious in view of the teaching of the prior art disclosed in applicant's specification. In other words, applicant argues that solving the problems associated with the prior art by itself presents indicia of unobviousness and applicant's claim would not have been obvious since it solves the prior art problem.

In addition, the distinction between the techniques used coupled with problem solutions are further evidence of unobviousness. Newly added Claims 75-84 are patentable for the reasons set forth above.

It is believed that the present amendment answers all of the issues raised by the examiner. Re-examination is hereby requested, and an early allowance of all the claims is solicited.

Attached hereto is a marked-up version of the changes made to the specification and claims by the current amendment. The attached page is captioned "Version with markings to show changes made".

Respectfully Submitted,

Joscelyn G. Cockburn

Joscelyn G. Cockburn, Reg. No. 27,069
Attorney of Record
IBM Corporation
Intellectual Property Law Dept. 9CCA/B002
P.O. Box 12195
Res. Tri. Park, NC 27709

JGC:ko
919-543-9036
FAX 919-254-2649

VERSION WITH MARKINGS TO SHOW CHANGES MADE

In the Specification:

Page 5, paragraph beginning at line 15 has been amended as follows:

-- Figure 8 is a block diagram of a table after column headings have been abbreviated ~~abbreviating~~ using a method in accordance with the present invention. –

In the Claims:

Claims 42, 45 and 47 have been amended as follows:

42. (Amended) A computer generated method for reducing an amount of horizontal space required when displaying a plurality of column on a display screen, at least one column of the plurality of columns including at least one entry containing text data, the method comprising the steps of:

(a) ~~determining a character type~~ defining a first character type as a space, a second character type as a lower case vowel and a third character type as a lower case letter;

(b) obtaining the at least one entry; and

(c) abbreviating a width of the at least one entry by

(c1) removing at least one ~~space~~ of the first character type if the at least one entry includes the at least one ~~space and if the character type is a~~ of the first character type~~, the first character type being a space~~; ~~and~~

(c2) removing at least one of the second character ~~having a second character~~ type if the at least one entry includes the at least one ~~character and if the~~ of the second character type ~~is the second character type~~; and (c3) removing at least one ~~lower case letter~~ of the third character type if the at least one entry includes the at least one ~~lower case letter and if~~ of the third character type~~ is a third character type, the third character type being a lower case letter~~.

45. (Amended) The method of claim 44 wherein the at least one column heading includes a first plurality of characters; and wherein abbreviating step (b) further includes the step of:

(b1) removing at least one character of a second plurality of characters if the at least one column heading includes the at least one character.

47. (Amended) The method of claim 44 wherein the at least one column heading includes a plurality of characters, the plurality of characters being capable of containing at least one lower case character not a lower case vowel; and wherein abbreviating step (b) further includes the step of:

(b1) removing the at least one lower case character if the at least one column heading includes the at least one lower case character.

Claims 75-84 have been added.